**大学生创新创业训练项目结题材料**

项目名称: 基于智能车路径规划算法的研究与实现

项目编号: 201710702054

项目负责人: 张彪 电话： 15025005925

指导教师: 吴琼 电话： 13630220923

所在学院: 计算机科学与工程学院

项目起止年月: 2017.5-2018.12.

装订日期： 2018年12月18日星期二

**西安工业大学**

**2018年 12月**

目录

[第一部分：《大学生创新训练项目申请书》 3](#_Toc533345677)

[第二部分：《西安工业大学创新创业训练计划项目合同书》 10](#_Toc533345678)

[第二部分：《西安工业大学大学生创新创业训练计划项目中期报告》 13](#_Toc533345679)

[第三部分：《西安工业大学大学生创新创业训练计划项目结题验收申请书》 32](#_Toc533345680)

[第四部分：获取专利 44](#_Toc533345681)

[第五部分：实验装置 50](#_Toc533345685)

[第六部分：软件说明书 52](#_Toc533345686)

[第七部分:设备操作说明书 59](#_Toc533345687)

## 第一部分：《大学生创新训练项目申请书》

大学生创新训练项目申请书

项目编号 201710702A052

项目名称 基于智能车路径规划算法的研究与实现

项目负责人 张彪 联系电话 **18392678200**

所在学院  **计算机科学与工程学院**

学 号 15040308118 专业班级 软件工程

指导教师 吴琼

E-mail wq0758@163.com

申请日期 2017.5.23

起止年月 2017.5-2018.5

西安工业大学

**填 写 说 明**

1、本申请书所列各项内容均须实事求是，认真填写，表达明确严谨，简明扼要

2、申请人可以是个人，也可为创新团队，首页只填负责人。“项目编号”一栏不填。

3、本申请书为大16开本（A4），左侧装订成册。可网上下载、自行复印或加页，但格式、内容、大小均须与原件一致。

4、负责人所在学院认真审核, 经初评和答辩，签署意见后，将申请书（一式两份）报送××××大学项目管理办公室。

* 1. 基本情况

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 项目  名称 | | 基于智能车路径规划算法的研究与实现 | | | | | | | | | | |
| 所属  学科 | | 计算机科学与技术 | | | | | | | | | | |
| 申请  金额 | | 15000 元 | | | | 起止年月 | | 2017年5月至 2018年 5月 | | | | |
| 负责人  姓名 | | 张彪 | | 性别 | | 男 | 民族 | 汉 | | 出生年月 | | 1998年5月 |
| 学号 | | 15040308118 | | 联系  电话 | | 宅：86173267 手机: 18392678200 | | | | | | |
| 指导  教师 | | 吴琼 | | 联系  电话 | | 宅： 86173267 手机:13630220923 | | | | | | |
| 负责人曾经参与科研的情况 | | | 无 | | | | | | | | | |
| 指导教师承担科研课题情况 | | | 1．2016年获批校长基金项目“基于智能车路径规划算法的研究与实现”；      2．2017年发表论文“智能机器人路径规划中的蚁群算法改进探讨”（统计员期刊）；      3．2017年发表论文“低数据量环境下物联网密钥管理算法与方案设计”（核心期刊）；      4．2015年获批实用新型专利“车辆充换电站路径规划系统”，2016年获批实用新型专利“一种基于车况和路况的智能车辆路径规划系统”； | | | | | | | | | |
| 指导教师对本项目的支持情况 | | | 负责任务的分工、负责设备的选用、负责相关指导工作、论文的修改发表等工作。 | | | | | | | | | |
| 项  目  组  主  要  成  员 | 姓 名 | | 学号 | | 专业班级 | | | | 所在学院 | | 项目中的分工 | |
| 李毛 | | 15100303114 | | 应用数学 | | | | 理学院 | | 算法分析 | |
| 李佳乐 | | 15100101130 | | 信息与计算科学 | | | | 理学院 | | 测试 | |
|  | |  | |  | | | |  | |  | |

* 1. 立项依据（可加页）

|  |
| --- |
| 1. **研究目的**     本项目主要是研究各种路径规划算法在智能车中的应用，并从获得的数据中比较各种算法的优缺点。   1. **研究内容**         根据机器人对环境信息感知的程度不同，将移动机器人路径规划方法分为两大类: 基于环境模型的全局路径规划方法和基于传感器信息的局部路径规划方法。目前，已有的全局路径规划算法有可视图法、A\*算法、栅格法、蚁群算法、遗传算法等，已有的局部路径规划算法有势场法、神经网络法和模糊逻辑法等。   1. **国、内外研究现状和发展动态**        目前国内外已经存在很多移动机器人路径规划问题的算法和应用，这是本课题研究的前提条件，但路径规划问题在智能车中的应用很少，并且各种算法之间效率的比较也基本没有。   1. **创新点与项目特色**         从各种路径规划算法中找到一种比较合适智能车使用的算法，为以后智能汽车的应用探索奠定基础。   1. **技术路线、拟解决的问题及预期成果**   技术路线、拟解决的问题：  1．路径规划算法的选择；  2．路径规划算法的实现；  3．不同路径规划算法的在同等条件下的效率分析；  4．路径规划算法的优化。  预期成果：  1. 智能小车一辆；  2. 相关篇论文；  3. 技术报告一份。​   1. **项目研究进度安排**   1．2017年4月-7月   选取路径规划算法、建模、编程程序；  2．2017年8月-10月   测试各中算法效率，分析数据，撰写论文；  3．2017年月11-12月  发表论文及相应成果申报；         4．2018年1月-4月   准备项目结题验收。   1. **已有基础**    1. **与本项目有关的研究积累和已取得的成绩**   本课题在前期已经做过相应的调研，购买了相应的书籍，并查阅了大量资料，购置了智能车等相应的硬件设备，设备的调试工作已经完成。     课题组成员具有一定的科研经验，成员结构和学科分布合理。   * 1. **已具备的条件，尚缺少的条件及解决方法**   对路径规划算法和智能车还不是很熟悉，进一步深入研究掌握相关知识。 |

* 1. 经费预算

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 开支科目 | 预算经费  （元） | 主要用途 | 阶段下达经费计划（元） | |
| 前半阶段 | 后半阶段 |
| 预算经费总额 | **15000** |  | **15000** | **0** |
| 1. 业务费 | **6500** | **相关费用** | **6500** | **0** |
| （1）计算、分析、测试费 | **0** |  | **0** | **0** |
| （2）能源动力费 | **0** |  | **0** | **0** |
| （3）会议、差旅费 | **3000** | **会议相关费用** | **3000** | **0** |
| （4）文献检索费 | **0** |  | **0** | **0** |
| （5）论文出版费 | **3500** | **论文发表** | **3500** | **0** |
| 2. 仪器设备购置费 | **6500** | **设备购置** | **6500** | **0** |
| 3. 实验装置试制费 | **2000** | **场地设置和测试费用** | **2000** | **0** |
| 4. 材料费 | **0** |  | **0** | **0** |
| 学校批准经费 | **15000** |  | **15000** | **0** |

* 1. 指导教师意见

|  |
| --- |
| 该题目涉及嵌入式技术，机械设计及运动控制等相关领域的知识，是一项综合性的交叉学科题目，对知识的综合应用有一定的难度。     智能两轮平衡车目前也是一个研究热点。     综上，该题目可行。    **导师（签章）：**  **年 月 日** |

* 1. 院系大学生创新创业训练计划专家组意见

|  |
| --- |
| 同意申报    **专家组组长（签章）：**  **年 月 日** |

* 1. 学校大学生创新创业训练计划专家组意见

|  |
| --- |
| 同意申报    **负责人（签章）：**  **年 月 日** |

* 1. 大学生创新创业训练计划领导小组审批意见

|  |
| --- |
| 同意申报    **负责人（签章）：**  **年 月 日** |

## 第二部分：《西安工业大学创新创业训练计划项目合同书》

西安工业大学

大学生创新创业训练计划项目

合同书

项目名称：**基于智能车路径规划算法的研究与实现＿**

项目编号：**＿201710702054＿＿＿＿＿**

项目负责人：**＿ 张彪＿＿＿＿＿**

项目启动日期：**＿2017年10月＿＿＿**

西安工业大学大学生科技创新基金委员会制

二0一七年十月

**西安工业大学大学生科技创新基金委员会（**以下简称甲方）

**项目负责人：** **张彪** （以下简称乙方）

　　为了促进大学生创新创业训练计划项目基金的有效使用，确保资助项目的实施，甲方与乙方特订立本合同。

　　一、项目名称：基于智能车路径规划算法的研究与实现

　　二、项目负责人：张彪

　　三、项目组其他成员：李毛、李佳

　　四、项目指导教师：吴琼

五、拟提供成果及成果形式：

（一）、结题基本要求

1、公开发表论文 0 篇。

（国家级项目要求至少发表一篇学术论文，且第一作者为学生；省级不做要求）

2、参加我校学科竞赛1项及以上，至少获得省级三等奖及以上0项（省级大创项目，对参与学科竞赛的获奖不做要求）。

盖章

（二）、结题技术要求

1、熟悉蚁群算法和A\*算法等路径规划算法；

2、根据算法建立模型并编写程序；

3、搭建测试环境，测试智能机器人；

4、分析测试数据；

5、撰写论文1篇；

6、撰写完成总结报告；

7、完成结题验收报告。

　　六、项目完成时间： 2018年10月 31日 。

　　七、资助经费金额：￥ 3000元。

　　八、乙方保证做到节约开支、专款专用，报销时经甲方审核。

　　九、乙方保证于2018年5月 10日前向甲方提交项目中期进度报告。

　　十、乙方保证于2018年10月31日前向甲方提交项目研究总结报告和发表的论文、成果实物及其证明材料。

十一、甲方按照《西安工业大学大学生创新创业训练计划项目管理办法》进行管理和监督，乙方按照《西安工业大学大学生创新创业训练计划项目立项申报表》填写的内容组织实施。

　　十二、本合同未尽事宜，应按学校的现行规定，经双方协商后作出补充规定。补充规定与本合同具有同等效力。

　　十三、本合同书除打印部分外，一律用蓝、黑签字笔或钢笔填写，未经合法授权代签或涂改无效。

　　十四、本合同一式三份，甲两份、乙方一份。

　　十五、本合同自双方签字之日起生效。

**甲方（教务处代章）** **乙方（签字并加盖学院行政章）**

项目负责人： 张彪

项目指导教师： 吴琼

教学院长： 刘白林

2017 年 10 月 22 日 2017年 10 月 17 日

备注：拟提供成果及成果形式要求严格按照项目原始申报书中的内容填写，不得删减。

## 第二部分：《西安工业大学大学生创新创业训练计划项目中期报告》

**西安工业大学大学生创新创业训练计划项目中期报告**

项目编号：201710702054 填写时间：2018 年 7月 10 日

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 项目名称 | 基于智能车路径规划算法的研究与实现 | | | 项目类别 | 省级 |
| 负责人 | 张彪 | 成员 | 李佳乐、李毛 | | |
| **一、项目进展情况**  1．A\*算法的c语言已经实现  A star算法介绍：  A\*（A-Star)算法是一种静态路网中求解最短路最有效的方法。  公式表示为： f(n)=g(n)+h(n), 其中f(n) 是节点n从初始点到目标点的估价函数，g(n) 是在状态空间中从初始节点到n节点的实际代价，h(n)是从n到目标节点最佳路径的估计代价。  保证找到最短路径（最优解的）条件，关键在于估价函数h(n)的选取：  估价值h(n)<= n到目标节点的距离实际值，这种情况下，搜索的点数多，搜索范围大，效率低。但能得到最优解。  如果估价值>实际值, 搜索的点数少，搜索范围小，效率高，但不能保证得到最优解。  估价值与实际值越接近，估价函数取得就越好。  例如对于几何路网来说，可以取两节点间欧几理德距离（直线距离）做为估价值，即f=g(n)+sqrt((dx-nx)\*(dx-nx)+(dy-ny)\*(dy-ny))；这样估价函数f在g值一定的情况下，会或多或少的受估价值h的制约，节点距目标点近，h值小，f值相对就小，能保证最短路的搜索向终点的方向进行。明显优于Dijstra算法的毫无无方向的向四周搜索。A\*算法流程图如图1所示。    图1 A\*算法流程  2. 在电机运转调试阶段  左右两边电机转速的控制上在进一步的探索，协调两边的转速，时间信号的控制上在进一步的调试中  3．现在进行超声波模块的使用的配置，以及超声波寻路算法结合到小车运行路径规划结合的框架的探索  超声波模块比如说  short Ultrasonic(void)  {  unsigned long ulTime;  unsigned long count;    if (cUltrasonicInit==0) Ultrasonic\_Init(); //设置GPIOA.6的中断和基本超声波环境的配置    switch(cTrigState) //根据状态调整  {  case IDLE:  WriteDO(TRIG,0);  cTrigState = TRIGSTART;  break;  case TRIGSTART:  WriteDO(TRIG,1);  cTrigState = TRIGEND;  break;  default:  cTrigState++;  break;  }  return sLength;  }  （可自行附页） | | | | | |
| **二、已取得的成果**  1．基本的A\*算法已经实现，并且在vs上运行成功，并且完成从vs到keil平台的移植  C代码实现：  #include <stdio.h>  #include <stdlib.h>    #define STARTNODE 1  #define ENDNODE 2  #define BARRIER 3    typedef struct AStarNode  {  int s\_x; // 坐标(最终输出路径需要)  int s\_y;  int s\_g; // 起点到此点的距离( 由g和h可以得到f，此处f省略，f=g+h )  int s\_h; // 启发函数预测的此点到终点的距离  int s\_style;// 结点类型：起始点，终点，障碍物  struct AStarNode \* s\_parent; // 父节点  int s\_is\_in\_closetable; // 是否在close表中  int s\_is\_in\_opentable; // 是否在open表中  }AStarNode, \*pAStarNode;    AStarNode map\_maze[10][10]; // 结点数组  pAStarNode open\_table[100]; // open表  pAStarNode close\_table[100]; // close表  int open\_node\_count; // open表中节点数量  int close\_node\_count; // close表中结点数量  pAStarNode path\_stack[100]; // 保存路径的栈  int top = -1; // 栈顶      // 交换两个元素  //  void swap( int idx1, int idx2 )  {  pAStarNode tmp = open\_table[idx1];  open\_table[idx1] = open\_table[idx2];  open\_table[idx2] = tmp;  }    // 堆调整  //  void adjust\_heap( int /\*i\*/nIndex )  {  int curr = nIndex;  int child = curr \* 2 + 1; // 得到左孩子idx( 下标从0开始，所有做孩子是curr\*2+1 )  int parent = ( curr - 1 ) / 2; // 得到双亲idx    if (nIndex < 0 || nIndex >= open\_node\_count)  {  return;  }  // 往下调整( 要比较左右孩子和cuur parent )  //  while ( child < open\_node\_count )  {  // 小根堆是双亲值小于孩子值  //  if ( child + 1 < open\_node\_count && open\_table[child]->s\_g + open\_table[child]->s\_h > open\_table[child+1]->s\_g + open\_table[child+1]->s\_h )  {  ++child; // 判断左右孩子大小  }    if (open\_table[curr]->s\_g + open\_table[curr]->s\_h <= open\_table[child]->s\_g + open\_table[child]->s\_h)  {  break;  }  else  {  swap( child, curr ); // 交换节点  curr = child; // 再判断当前孩子节点  child = curr \* 2 + 1; // 再判断左孩子  }  }    if (curr != nIndex)  {  return;  }    // 往上调整( 只需要比较cuur child和parent )  //  while (curr != 0)  {  if (open\_table[curr]->s\_g + open\_table[curr]->s\_h >= open\_table[parent]->s\_g + open\_table[parent]->s\_h)  {  break;  }  else  {  swap( curr, parent );  curr = parent;  parent = (curr-1)/2;  }  }  }    // 判断邻居点是否可以进入open表  //  void insert\_to\_opentable( int x, int y, pAStarNode curr\_node, pAStarNode end\_node, int w )  {  int i;    if ( map\_maze[x][y].s\_style != BARRIER ) // 不是障碍物  {  if ( !map\_maze[x][y].s\_is\_in\_closetable ) // 不在闭表中  {  if ( map\_maze[x][y].s\_is\_in\_opentable ) // 在open表中  {  // 需要判断是否是一条更优化的路径  //  if ( map\_maze[x][y].s\_g > curr\_node->s\_g + w ) // 如果更优化  {  map\_maze[x][y].s\_g = curr\_node->s\_g + w;  map\_maze[x][y].s\_parent = curr\_node;    for ( i = 0; i < open\_node\_count; ++i )  {  if ( open\_table[i]->s\_x == map\_maze[x][y].s\_x && open\_table[i]->s\_y == map\_maze[x][y].s\_y )  {  break;  }  }    adjust\_heap( i ); // 下面调整点  }  }  else // 不在open中  {  map\_maze[x][y].s\_g = curr\_node->s\_g + w;  map\_maze[x][y].s\_h = abs(end\_node->s\_x - x ) + abs(end\_node->s\_y - y);  map\_maze[x][y].s\_parent = curr\_node;  map\_maze[x][y].s\_is\_in\_opentable = 1;  open\_table[open\_node\_count++] = &(map\_maze[x][y]);  }  }  }  }    // 查找邻居  // 对上下左右8个邻居进行查找  //  void get\_neighbors( pAStarNode curr\_node, pAStarNode end\_node )  {  int x = curr\_node->s\_x;  int y = curr\_node->s\_y;    // 下面对于8个邻居进行处理！  //  if ( ( x + 1 ) >= 0 && ( x + 1 ) < 10 && y >= 0 && y < 10 )  {  insert\_to\_opentable( x+1, y, curr\_node, end\_node, 10 );  }    if ( ( x - 1 ) >= 0 && ( x - 1 ) < 10 && y >= 0 && y < 10 )  {  insert\_to\_opentable( x-1, y, curr\_node, end\_node, 10 );  }    if ( x >= 0 && x < 10 && ( y + 1 ) >= 0 && ( y + 1 ) < 10 )  {  insert\_to\_opentable( x, y+1, curr\_node, end\_node, 10 );  }    if ( x >= 0 && x < 10 && ( y - 1 ) >= 0 && ( y - 1 ) < 10 )  {  insert\_to\_opentable( x, y-1, curr\_node, end\_node, 10 );  }    if ( ( x + 1 ) >= 0 && ( x + 1 ) < 10 && ( y + 1 ) >= 0 && ( y + 1 ) < 10 )  {  insert\_to\_opentable( x+1, y+1, curr\_node, end\_node, 14 );  }    if ( ( x + 1 ) >= 0 && ( x + 1 ) < 10 && ( y - 1 ) >= 0 && ( y - 1 ) < 10 )  {  insert\_to\_opentable( x+1, y-1, curr\_node, end\_node, 14 );  }    if ( ( x - 1 ) >= 0 && ( x - 1 ) < 10 && ( y + 1 ) >= 0 && ( y + 1 ) < 10 )  {  insert\_to\_opentable( x-1, y+1, curr\_node, end\_node, 14 );  }    if ( ( x - 1 ) >= 0 && ( x - 1 ) < 10 && ( y - 1 ) >= 0 && ( y - 1 ) < 10 )  {  insert\_to\_opentable( x-1, y-1, curr\_node, end\_node, 14 );  }  }      int main()  {  // 地图数组的定义  //  AStarNode \*start\_node; // 起始点  AStarNode \*end\_node; // 结束点  AStarNode \*curr\_node; // 当前点  int is\_found; // 是否找到路径  int maze[][10] ={ // 仅仅为了好赋值给map\_maze  { 1,0,0,3,0,3,0,0,0,0 },  { 0,0,3,0,0,3,0,3,0,3 },  { 3,0,0,0,0,3,3,3,0,3 },  { 3,0,3,0,0,0,0,0,0,3 },  { 3,0,0,0,0,3,0,0,0,3 },  { 3,0,0,3,0,0,0,3,0,3 },  { 3,0,0,0,0,3,3,0,0,0 },  { 0,0,2,0,0,0,0,0,0,0 },  { 3,3,3,0,0,3,0,3,0,3 },  { 3,0,0,0,0,3,3,3,0,3 },  };  int i,j,x;    // 下面准备点  //  for( i = 0; i < 10; ++i )  {  for ( j = 0; j < 10; ++j )  {  map\_maze[i][j].s\_g = 0;  map\_maze[i][j].s\_h = 0;  map\_maze[i][j].s\_is\_in\_closetable = 0;  map\_maze[i][j].s\_is\_in\_opentable = 0;  map\_maze[i][j].s\_style = maze[i][j];  map\_maze[i][j].s\_x = i;  map\_maze[i][j].s\_y = j;  map\_maze[i][j].s\_parent = NULL;    if ( map\_maze[i][j].s\_style == STARTNODE ) // 起点  {  start\_node = &(map\_maze[i][j]);  }  else if( map\_maze[i][j].s\_style == ENDNODE ) // 终点  {  end\_node = &(map\_maze[i][j]);  }    printf("%d ", maze[i][j]);  }    printf("\n");  }    // 下面使用A\*算法得到路径  //  open\_table[open\_node\_count++] = start\_node; // 起始点加入open表    start\_node->s\_is\_in\_opentable = 1; // 加入open表  start\_node->s\_g = 0;  start\_node->s\_h = abs(end\_node->s\_x - start\_node->s\_x) + abs(end\_node->s\_y - start\_node->s\_y);  start\_node->s\_parent = NULL;    if ( start\_node->s\_x == end\_node->s\_x && start\_node->s\_y == end\_node->s\_y )  {  printf("起点==终点！\n");  return 0;  }    is\_found = 0;    while( 1 )  {  // for test  //  /\* for ( x = 0; x < open\_node\_count; ++x )  {  printf("(%d,%d):%d ", open\_table[x]->s\_x, open\_table[x]->s\_y, open\_table[x]->s\_g+open\_table[x]->s\_h);  }  printf("\n\n");  \*/  curr\_node = open\_table[0]; // open表的第一个点一定是f值最小的点(通过堆排序得到的)  open\_table[0] = open\_table[--open\_node\_count]; // 最后一个点放到第一个点，然后进行堆调整  adjust\_heap( 0 ); // 调整堆    close\_table[close\_node\_count++] = curr\_node; // 当前点加入close表  curr\_node->s\_is\_in\_closetable = 1; // 已经在close表中了    if ( curr\_node->s\_x == end\_node->s\_x && curr\_node->s\_y == end\_node->s\_y )// 终点在close中，结束  {  is\_found = 1;  break;  }    get\_neighbors( curr\_node, end\_node ); // 对邻居的处理    if ( open\_node\_count == 0 ) // 没有路径到达  {  is\_found = 0;  break;  }  }    if ( is\_found )  {  curr\_node = end\_node;    while( curr\_node )  {  path\_stack[++top] = curr\_node;  curr\_node = curr\_node->s\_parent;  }    while( top >= 0 ) // 下面是输出路径看看~  {  if ( top > 0 )  {  printf("(%d,%d)-->", path\_stack[top]->s\_x, path\_stack[top--]->s\_y);  }  else  {  printf("(%d,%d)", path\_stack[top]->s\_x, path\_stack[top--]->s\_y);  }  }  }  else  {  printf("么有找到路径");  }    puts("");    return 0;  }    运行结果    2．Stm32平台的GPIO的使用已经配置好，PWM以及调整电机转速的程序框架已经完成，现阶段在调试阶段。  Main.c  #include "SteeringMotor.h"  #include "Model.h"  #include "delay.h"  /////////////////////////////////////////////    /////////////////////////////////////////////  #define PIT 5  void Initial(void)  {  SysTick\_Init(72);//设置系统时钟  TIM4\_PWM\_Init(7199,0);  SetMotorSpeed(1,0);//设置左轮速度  SetMotorSpeed(0,0);//设置右轮速度  delay\_ms(500); //设置延迟时间0.5秒  }  int main(void)  {  Initial();  while (1)  {  SetMotorSpeed(1,50);  SetMotorSpeed(0,50);  delay\_ms(3000);  SetMotorSpeed(1,-50);  SetMotorSpeed(0,-50);  delay\_ms(3000);    }  }  //设置电机驱动程序的头文件  SteeringMotor.h  #ifndef \_\_PWM\_H  #define \_\_PWM\_H  #include "stm32f10x\_gpio.h"  #include "stm32f10x\_tim.h"  #include "stm32f10x\_rcc.h"  void TIM1\_PWM\_Init(unsigned short arr,unsigned short psc);  void TIM4\_PWM\_Init(unsigned short arr,unsigned short psc);  void SteeringMotorInitial(void);  void SetSteeringMotor(short dSpeed);  void SetMotorSpeed(unsigned char ucChannel,signed char cSpeed);  void SetMotor\_Speed(unsigned char ucChannel,signed char cSpeed);  void MotorControl(void);  #endif  //实现电机控制的函数实现  SteeringMotor.c  #include "SteeringMotor.h"  #include "delay.h"  #include "IO.h"  #include "Math.h"  #include "REG.h"  signed short sPWMR,sPWML,dPWM;  //PWM输出初始化  //arr：自动重装值  //psc：时钟预分频数  //struct SPin DMPIN = {GPIOA, GPIO\_Pin\_8,GPIO\_Mode\_AF\_PP,GPIO\_Speed\_50MHz};  struct SPin M1DIR = {GPIOB, GPIO\_Pin\_2,GPIO\_Mode\_Out\_PP,GPIO\_Speed\_50MHz};  struct SPin M1EN = {GPIOB, GPIO\_Pin\_8,GPIO\_Mode\_AF\_PP,GPIO\_Speed\_50MHz};  struct SPin M2DIR = {GPIOB, GPIO\_Pin\_3,GPIO\_Mode\_Out\_PP,GPIO\_Speed\_50MHz};  struct SPin M2EN = {GPIOB, GPIO\_Pin\_9,GPIO\_Mode\_AF\_PP,GPIO\_Speed\_50MHz};  void TIM4\_PWM\_Init(unsigned short arr,unsigned short psc)  {  TIM\_TimeBaseInitTypeDef TIM\_TimeBaseStructure;  TIM\_OCInitTypeDef TIM\_OCInitStructure;    RCC\_APB1PeriphClockCmd(RCC\_APB1Periph\_TIM4, ENABLE);//  RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_GPIOB , ENABLE); //使能GPIO  外设时钟使用    DIO\_Initial(M2EN);  DIO\_Initial(M2DIR);  DIO\_Initial(M1EN);  DIO\_Initial(M1DIR);  TIM\_TimeBaseStructure.TIM\_Period = arr; //设置在下一个更新事件装入活动的自动重装载寄存器周期的值 80K  TIM\_TimeBaseStructure.TIM\_Prescaler =psc; //设置用来作为TIMx时钟频率除数的预分频值 不分频  TIM\_TimeBaseStructure.TIM\_ClockDivision = 0; //设置时钟分割:TDTS = Tck\_tim  TIM\_TimeBaseStructure.TIM\_CounterMode = TIM\_CounterMode\_Up; //TIM向上计数模式  TIM\_TimeBaseInit(TIM4, &TIM\_TimeBaseStructure); //根据TIM\_TimeBaseInitStruct中指定的参数初始化TIMx的时间基数单位    TIM\_OCInitStructure.TIM\_OCMode = TIM\_OCMode\_PWM2; //选择定时器模式:TIM脉冲宽度调制模式2  TIM\_OCInitStructure.TIM\_OutputState = TIM\_OutputState\_Enable; //比较输出使能  TIM\_OCInitStructure.TIM\_Pulse = 0; //设置待装入捕获比较寄存器的脉冲值  TIM\_OCInitStructure.TIM\_OCPolarity = TIM\_OCPolarity\_High; //输出极性:TIM输出比较极性高  TIM\_OC3Init(TIM4, &TIM\_OCInitStructure); //根据TIM\_OCInitStruct中指定的参数初始化外设TIMx  TIM\_OC4Init(TIM4, &TIM\_OCInitStructure); //根据TIM\_OCInitStruct中指定的参数初始化外设TIMx  TIM\_CtrlPWMOutputs(TIM4,ENABLE); //MOE 主输出使能  TIM\_OC3PreloadConfig(TIM4, TIM\_OCPreload\_Enable); //CH1预装载使能  TIM\_OC4PreloadConfig(TIM4, TIM\_OCPreload\_Enable); //CH1预装载使能  TIM\_ARRPreloadConfig(TIM4, ENABLE); //使能TIMx在ARR上的预装载寄存器  TIM\_Cmd(TIM4, ENABLE); //使能TIM1  }  void SetMotorSpeed(unsigned char ucChannel,signed char cSpeed)  {  static short sMotorSpeed = 0;  short sPWM;  float fDir = 1;    if (cSpeed>=100) cSpeed = 100;  if (cSpeed<=-100) cSpeed = -100;  sPWM = 7201 - fabs(cSpeed)\*72;  switch(ucChannel)  {  case 0://右轮  TIM\_SetCompare3(TIM4,sPWM);  if (cSpeed>0)  WriteDO(M1DIR,0);  else if(cSpeed<0)  WriteDO(M1DIR,1);  break;  case 1://左轮  TIM\_SetCompare4(TIM4,sPWM);  if (cSpeed>0)  WriteDO(M2DIR,1);  else if (cSpeed<0)  WriteDO(M2DIR,0);  break;  }  }  signed char cObjSpeed[2]={0,0};  void SetMotor\_Speed(unsigned char ucChannel,signed char cSpeed)  {  if (ucChannel>1) return;  if (cSpeed>=100) cSpeed = 100;  if (cSpeed<=-100) cSpeed = -100;  cObjSpeed[ucChannel]=cSpeed;  }  void MotorControl(void)  {  static signed char cCurrentSpeed[2]={0,0};  signed char cStep=5,i;  for (i=0;i<2;i++)  {  if (cObjSpeed[i]>(cCurrentSpeed[i]+cStep)) cCurrentSpeed[i]+=cStep;  else if (cObjSpeed[i]<(cCurrentSpeed[i]-cStep)) cCurrentSpeed[i]-=cStep;  else cCurrentSpeed[i]=cObjSpeed[i];  SetMotorSpeed(i,cCurrentSpeed[i]);  }  }  Reg.h  #ifndef \_\_AHRSREG\_H  #define \_\_AHRSREG\_H  #define REGRETURNSIZE 20  #define REGSIZE 0x40  #ifdef REG  short sReg[REGSIZE];  #else  extern signed short sReg[REGSIZE];  #endif  //Reg Name Define  //Public Reg, Do Not Change  #define SAVE 0x00  #define CALSW 0x01  #define RSW 0x02  #define RRATE 0x03  #define BAUD 0x04  #define RUNMODE 0x05  #define READADDR 0x06  //Private Reg, User Define  #define MOTERPWM 0x07  #define DMPWM 0x08  #define VSCALE 0x09  #define IRL 0x0a  #define IRR 0x0b  #define BWIRL 0x0c  #define BWIRR 0x0d  #define VOTAGE 0x0e  #define ENCODEV 0x0f  #define ULTRONIC 0x10  #define TEMP 0x11  #define HUMIDITY 0x12  #define AX 0x13  #define AZ 0x14  #define AY 0x15  #define WX 0x16  #define WY 0x17  #define WZ 0x18  #define ANGLE\_X 0x19  #define ANGLE\_Y 0x1a  #define ANGLE\_Z 0x1b  #define LatL 0x1c  #define LatH 0x1d  #define LonL 0x1e  #define LonH 0x1f  #define GPSHeight 0x20  #define GPSYAW 0x21  #define SVNUM 0x22  #define GPSVL 0x23  #define GPSVH 0x24  #define PDOP 0x25  #define HDOP 0x26  #define VDOP 0x27  #define YYMM 0x28  #define DDHH 0x29  #define MMSS 0x2a  #define MS 0x2b  #define GPSBAUD 0x2c  #define IRLMAX 0x2d  #define IRLMIN 0x2e  #define IRRMAX 0x2f  #define IRRMIN 0x30  #define BWLTHR 0x31  #define BWRTHR 0x32  #define RSV 0x33  //Run Mode Define  #define ULTRASONIC 0x00  #define INFRARED 0x01  #define TRACK 0x02  #define FOLLOW 0x03  #define GUESTURE 0x04  #define MOTION 0x05  #define IDLE 0x06  #define STOP 0x07  //Feadback Content  #define TIME 0x50  #define ACC 0x51  #define GYRO 0x52  #define ANGLE 0x53  #define MAGNETIC 0x54  #define PORT 0x55  #define PRESS 0x56  #define LONLAT 0x57  #define VELOCITY 0x58  #define REGVALUE 0x5F    void RegUpdate(unsigned char ucID,short sData);  void CheckRegUpdate(void);  void CheckRegReturn(void);  void ReturnRegData(unsigned char ucRegType ,unsigned char ucStartReg);  void RegAck(unsigned char ucRegID);  void RegInit(void);  #endif  调试代码  3．发表了1篇论文  （可自行附页） | | | | | |
| 三、经费开支情况  1．论文版面费  2．专利申报费用  3．实验相关耗材  （可自行附页） | | | | | |
| 四、存在的问题和困难  1．由于蚁群算法难度偏大，项目进度较为迟缓，就是stm32平台的一些使用，库函数的配置上有些困难，对于smt32整体没有一个整体的认识；  2．硬件平台编程需要继续熟悉，对于有些bug还是查不出来原因，网上也找不到。  （可自行附页） | | | | | |
| 五、建议和要求  需要深入学习stm32编程的技术 | | | | | |
| 1. 指导教师意见   签字 | | | | | |
| 七、学院意见  **部门盖章（主管领导签名）：** 年 月 日 | | | | | |
| 八、学校意见  领导签名（盖章）： 年 月 日 | | | | | |

说明：

中期报告电子版文件名命名方式：项目编号中期.doc

例：201710702006中期.doc；

|  |  |  |
| --- | --- | --- |
| **项目编码** | **项目名称** | **项目负责人** |
| 201710702006 | 基于嵌入式的车载稳定平台控制系统设计 | 张智勇 |
| 201710702010 | 基于微信公众号（小程序）的校内二手书交易和校内跑腿平台 | 田浩帅 |
| 201710702015 | 基于Scrapoy的分布式网络爬虫系统的设计与实现 | 冯树荣 |
| 201710702016 | 基于python的图灵机器人 | 陈伟 |
| 201710702038 | 面向“互联网+”的校园慈善基金会的建设与商业运营 | 李腾 |
| 201710702048 | 基于Windows的网络攻击智能防御系统 | 张卫博 |
| 201710702054 | 基于智能车路径规划算法的研究与实现 | 张彪 |
| 201710702063 | 基于敏感信息挖掘的网络舆情监测系统 | 石雷刚 |
| 201710702071 | 入侵检测系统和防火墙联动的校园网安全平台设计 | 高安兵 |
| 201710702075 | 教育头条—基于大数据分析的教育新闻网站开发 | 方乐 |
| 201710702079 | 智能喷淋养生监控系统的设计与实现 | 赵梦妮 |
| 201710702089 | 货车燃油及蓄电池安防一体检测系统 | 董永博 |
| 201710702101 | 基于Android的瓶盖识别与计数技术研究 | 郑思林 |
| 201710702108 | 基于Android的工业燃气锅炉物联网系统的设计 | 牛卓 |
| 201710702111 | 无人机复杂地形降落装置 | 王浩同 |

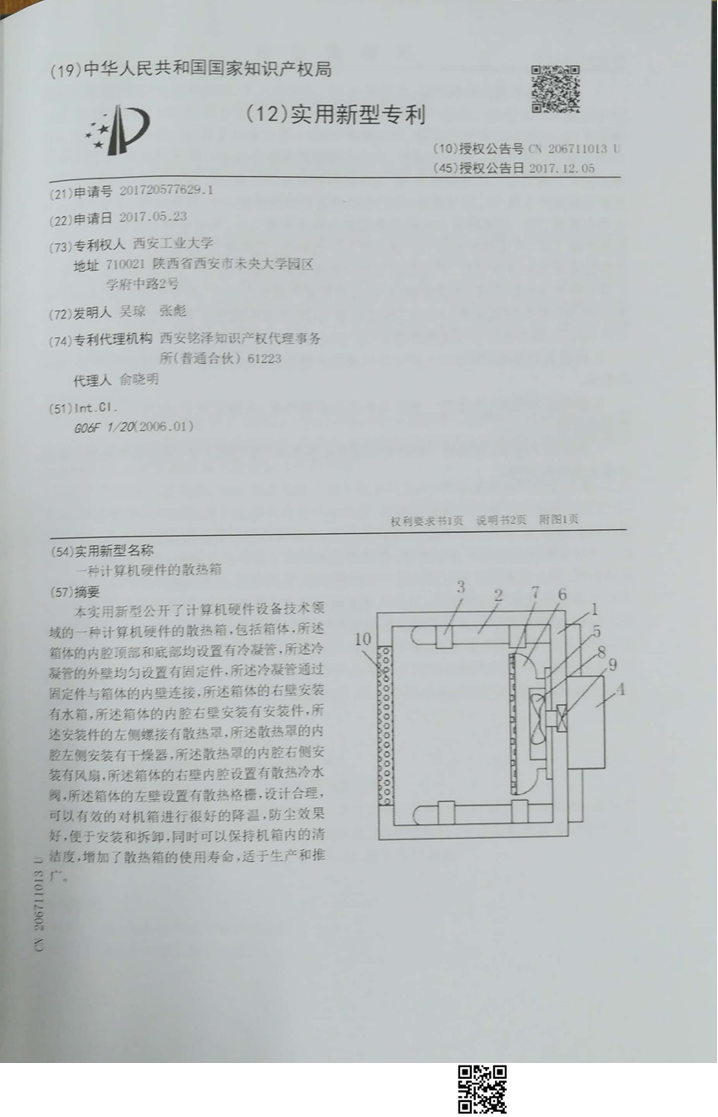
## 第三部分：《西安工业大学大学生创新创业训练计划项目结题验收申请书》

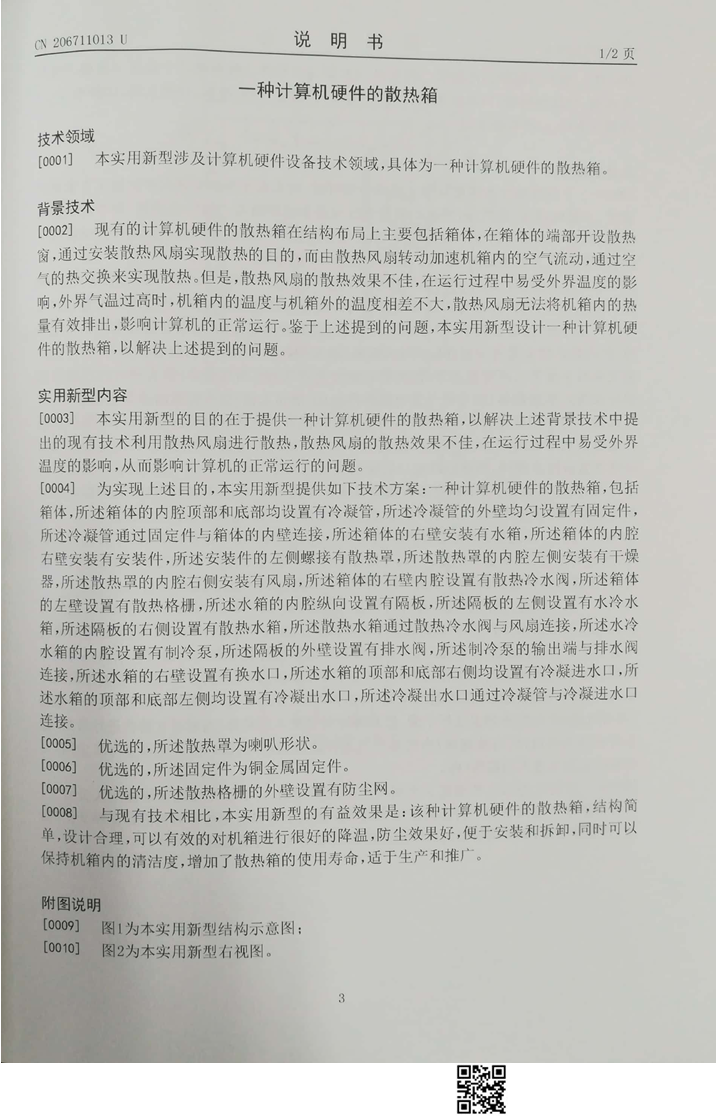
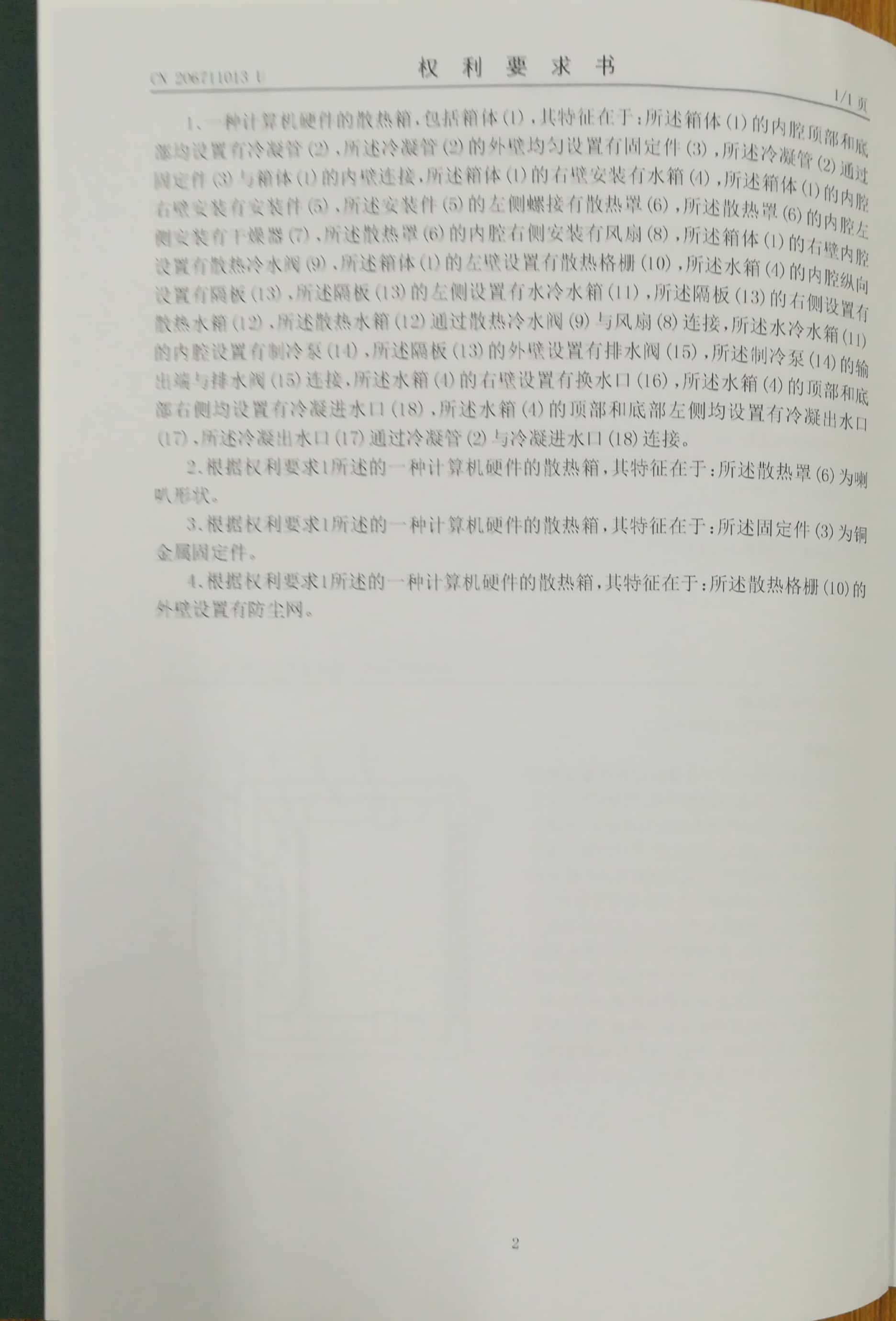
**西安工业大学大学生创新创业训练计划项目结题验收申请书**

项目编号：201710702054填写时间：2018 年10 月22 日

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 项目名称 | 基于智能车路径规划算法的研究与实现 | | | 项目类别 | 省级 |
| 负责人 | 张彪 | 主要成员 | 李佳乐、李毛 | | |
| 所属学院 | 计算机科学与工程学院 | 指导教师 | 吴琼 | | |
|  | 1. 项目完成情况 2. 算法方面，我们对于复杂，难以在当前水平下实现的蚁群，神经网络算法使用matlab进行软件仿真。   **蚁群算法**，在使用蚁群算法求解现实问题时，先生成具有一定数量蚂蚁的蚁群，让每一只蚂蚁建立一个解或解的一部分，每只人工蚁从问题的初始状态出发，根据“激素”浓度来选择下一个要转移到的状态，直到建立起一个解，每只蚂蚁根据所找到的解的好坏程度在所经过的状态上释放与解的质量成正比例的“信息素”。之后，每只蚂蚁又开始新的求解过程，直到寻找到满意解  è¿éåå¾çæè¿°  è¿éåå¾çæè¿°  蚁个体间通过这种信息交流机制来搜索食物。而用来交流反馈的化学因素现在被我们称之为——————“信息素”。  建立相关“双桥”实验的数学模型，首先，假设在对称桥的信息素的总数与过去一段时间内经过该桥的蚂蚁数目成正比（即每只蚂蚁都具有相同的信息素释放能力）；其次，假设某时刻蚂蚁按照桥上残留信息量的多少来选择其中的某条路径，经过该路径的蚂蚁数目越多，则该桥上的残留信息素总量就越大。    蚁群算法流程图    **深度优先搜索算法**  在搜索过程中访问某个顶点后，需要递归地访问此顶点的所有未访问过的相邻顶点。 初始条件下所有节点为白色，选择一个作为起始顶点，按照如下步骤遍历： a. 选择起始顶点涂成灰色，表示还未访问 b. 从该顶点的邻接顶点中选择一个，继续这个过程（即再寻找邻接结点的邻接结点），一直深入下去，直到一个顶点没有邻接结点了，涂黑它，表示访问过了 c. 回溯到这个涂黑顶点的上一层顶点，再找这个上一层顶点的其余邻接结点，继续如上操作，如果所有邻接结点往下都访问过了，就把自己涂黑，再回溯到更上一层。 d. 上一层继续做如上操作，知道所有顶点都访问过  http://my.csdn.net/uploads/201205/12/1336824807_2219.png  深度优先按照以下的顺序来搜索：  1.V0->V1->V4，此时到底尽头，仍然到不了V6，于是原路返回到V1去搜索其他路径；  2.返回到V1后既搜索V2，于是搜索路径是V0->V1->V2->V6,，找到目标节点，返回有解。  这样搜索只是2步就到达了，但是如果用BFS的话就需要多几步  广度优先搜索  广度优先搜索在进一步遍历图中顶点之前，先访问当前顶点的所有邻接结点。 a .首先选择一个顶点作为起始结点，并将其染成灰色，其余结点为白色。 b. 将起始结点放入队列中。 c. 从队列首部选出一个顶点，并找出所有与之邻接的结点，将找到的邻接结点放入队列尾部，将已访问过结点涂成黑色，没访问过的结点是白色。如果顶点的颜色是灰色，表示已经发现并且放入了队列，如果顶点的颜色是白色，表示还没有发现 d. 按照同样的方法处理队列中的下一个结点。 基本就是出队的顶点变成黑色，在队列里的是灰色，还没入队的是白色  基本流程图如下：  http://my.csdn.net/uploads/201204/30/1335725885_9403.png  举例        从顶点1开始进行广度优先搜索：   1. 初始状态，从顶点1开始，队列={1} 2. 访问1的邻接顶点，1出队变黑，2,3入队，队列={2,3,} 3. 访问2的邻接结点，2出队，4入队，队列={3,4} 4. 访问3的邻接结点，3出队，队列={4} 5. 访问4的邻接结点，4出队，队列={ 空} 结点5对于1来说不可达   对于相对有比较简单的**深度优先搜索**和广度优先搜索算法，我们使用devc++先是进行了软件上的逻辑仿真，之后又在硬件平台上结合红外线模块对于障碍物的第一步判断，如果一个方向方向有障碍物，则调整内轮转速变小，外轮不变，转向90度，对于当前位置的其他方向进行判断，如果可行，则设置或者保留当前位置，继续向可行方向走；如果不行，则退到上一次可行的位置，转一个角度（每次转一个角度都记录一个数字，防止原地打转，四次之内如果没方向可走，证明死路，小车停在原地即可）。  **深度优先算法**    对于A star算法，我们还是现在dev c++上实现基础的思路：它是启发式搜索，需要维护一个open,close表，我们在实现的时候是这么设计的：现在小车在一个位置，给小车一个目标位置，小车先在原点先原地“转”一圈，每次转45度，用于收集小车八个方向距离的期望最小值，距离期望值由超声波发射器对于周边的障碍取以及当前位置和目标位置的距离取最小值（堆排序）获得，最小值所在的方向被加入到open表中，作为小车下一个方向的指导，小车在走到新位置后，将上一个位置的信息放入到closed表中，一方面是不让小车返回来，另一方面减少搜索方向，加快效率。如此最终小车按照最优的距离最小值的指导来到目标地点，达到算法和路径上的最优。  **A star算法**     1. 硬件平台的解耦已经完成，并将算法写入到小车中，并经过了测试   硬件平台的编程与实现主要由负责人进行，在初期算法的与小车模型上的建立和实现上遇到了巨大的困难，经各类资料的查询与学习后，我们终于克服了代码的实现以及算法与小车之间的模型的建立，基本将a star算法，深度优先算法移植到了智能小车上，并反复测试，逐个解决了小车不按照代码逻辑运行，小车两个轮子转速不匹配，小车转弯角度的控制不正确等等问题。现在小车基本可以按照预期的路径规划算法(a star)算法运行。 | | | | |
|  | 二、已取得的成果  1.按照预期目标，我们在不断学习与实践中实现了基于智能车的路径规划算法包括**深度优先寻路算法、广度优先寻路算法、蚁群算法，A star算法**的学习，配合与现实中一些实际情况与算法的结合展开研究与设计，争取设计出一个适合我们自己的，在硬件小车上运行路径算法的模型，，在这个期间我们查阅和学习了大量由于深度学习，路径规划，智能机器人寻路的一些算法和论文，在这个过程的不断积累和讨论下，**我们设计出适合我们的，易于大家理解和实现的模型，并且在一次又一次的调试中实现了基于智能车的路径规划算法模型；**  **Matlab模拟bp神经网络**     1. 第二个方面，我们不仅在算法算法学习了很多课堂以外的东西，我们还学到了有关于嵌入式编程的东西，包括**stm32平台的编程基础，硬件架构，互联网底层驱动，串口**等与计算机底层有关联的东西，加强了对计算机体系的理解与体会;   Stm32架构    Stm32编程部分截图      3.第三个方面，由于我们在算法的模拟方面需要有大量的数学模型的建立和抽象，所以，我们另外还**学习了matlab**这样的数学软件，结合我们的路径规划算法，在matlab环境下对**神经网络机器人以及蚁群算法等高级统计算法**进行数学仿真，加强了对于这一类启发式的，人工智能算法的学习和理解。  4.在指导老师课题开展的同时，我们还协助指导老师一起查阅资料，深入理解物联网技术应用前景进行调查，并**发表了1篇论文**《物联网技术在智能家居中应用》，**并获批1项使用新型专利**。      （可自行附页） | | | | |
|  | 三、经费开支情况  总经费：0.4万   |  |  |  | | --- | --- | --- | | 经费支出项目 | 金额 | 备注 | | 论文发表 | 2000 | 版面费 | | 专利申请 | 1800 | 专利申请费用 | | 设备购置 | 260 | 智能小车 | | 办公耗材 |  |  | | 交通费 |  |  | | 其他 |  |  | | 经费合计 |  |  |   （可自行附页） | | | | |
|  | 四、指导教师意见  签字： | | | | |
|  | 五、学院意见  部门盖章（主管领导签名）： 年 月 日 | | | | |
|  | 六、大学生科技创新基金委员会意见  签字：（盖章） 年 月 日 | | | | |

## 第四部分：获取专利





## 

## 第五部分：实验装置

使用contex3内核的，带有超声波测距的stm32架构的智能小车

STM32与单片机

补充：

STM32是一种功能比较强大的32位的单片机。它和8位单片机最大的不同是，它不仅可以使用寄存器进行编程，还可以使用官方提供的库文件进行编程，这样不仅编程方便，而且更容易移植。

另外由于嵌入式系统主要分为DSP，微控制器MCU，微处理器，SOC四种。并且一个完整的嵌入式系统，从开发过程分为4层：硬件层、驱动层、操作系统层和应用层。而我们使用的STM32是MCU的一种。Stm32的设计人员将原来的嵌入式编程进行封装，得到了STM32平台上的编程库。一系列的内层驱动都被封装成了一系列的结构体。极大简化了开发以及工程化的管理。

如图：

如图为我们的设备，前面两个大的器件是超声波模块，我们使用的是

C-SR04超声波测距模块可提供2cm-400cm的非接触式距离感测功能，测距精度可达高到3mm；模块包括超声波发射器、接收器与控制电路。  
基本工作原理：  
(1)采用IO口TRIG触发测距，给最少10us的高电平信呈。  
(2)模块自动发送8个40khz的方波，自动检测是否有信号返回；  
(3)有信号返回， 通过IO口ECHO输出一个高电平， 高电平持续的时间就是超声波从发射到返回的时间。 测试距离=(高电平时间\*声速(340M/S))/2;

依赖平台：keil5

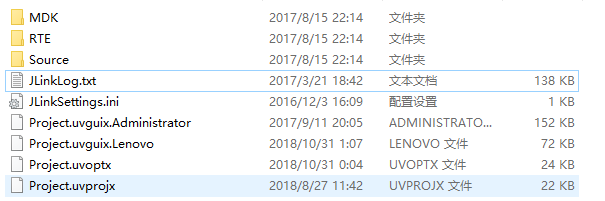
Keil MDK-ARM是美国Keil软件公司（现已被ARM公司收购）出品的支持ARM微控制器的一款IDE（集成开发环境）。MDK-ARM包含了工业标准的Keil C编译器、宏汇编器、调试器、实时内核等组件。具有业行领先的ARM C/C++编译工具链，完美支持Cortex-M、Cortex-R4、ARM7和ARM9系列器件，包含世界上品牌的芯片。比如：ST、Atmel、Freescale、NXP、TI等众多大公司微控制器芯片。像我们的stm32就使用的是cortex-M或者R3或者R4的架构

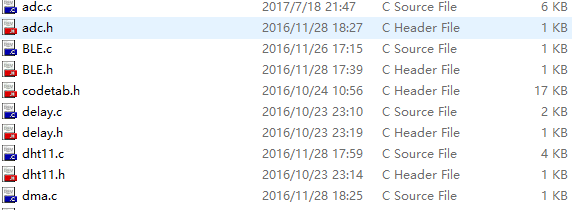
## 第六部分：软件说明书

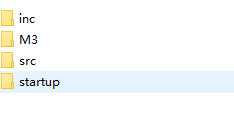
整个项目的整体框架有两部分组成，一部分是来自于STM32库，

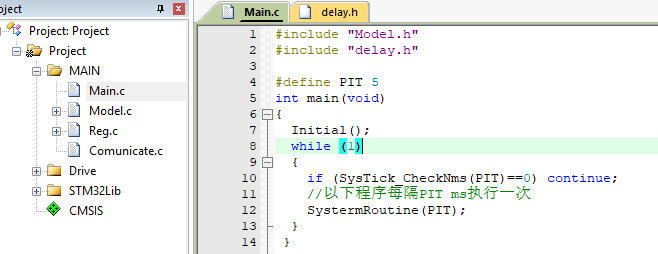
如图：

一个工程文件如下：包含Keil5工程任务键，MDK组件，以及运行时组件RET,以及项目源码文件Source，Source文件由STM32库文件

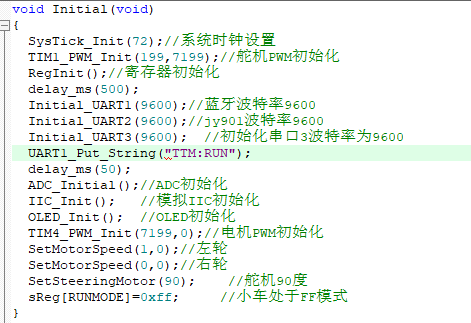


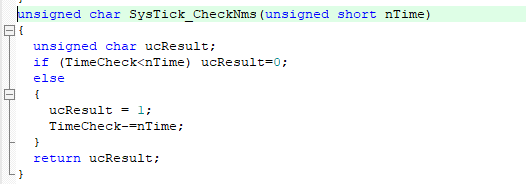
STM32库由STM32F10xlib系统库以及各部分，比如GPIO，I2C，串口的库文件都在驱动Drive目录下

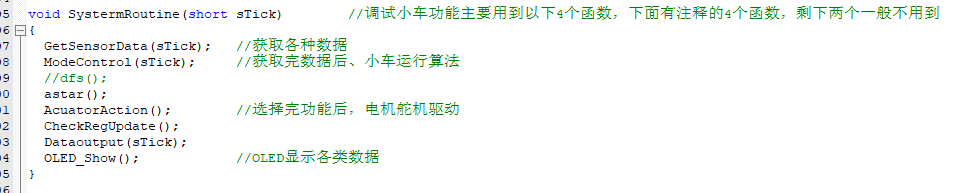
，App是我们的配置和运行自定义程序的地方，



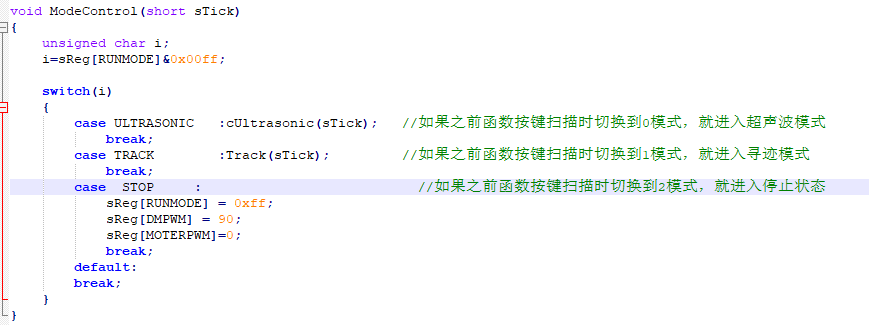
Main程序是程序开始执行的地方，使用Initial函数做系统环境的初始化，一般顺序是先初始化系统时钟频率，再初始化各个模块，整个流程

初始化模块，初始化系统时钟，舵机的PWM，各个功能寄存器，蓝牙，串口，adc,I2C,以及电机的PWM等的设置，并将小车至于FF模式，默认为超声波避障模式



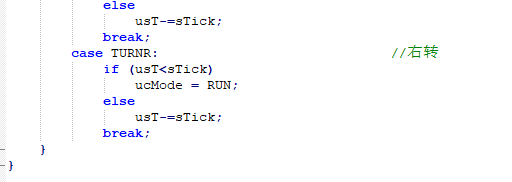
检测更新时间每五秒更新一次，就是大概的一个外部控制，用于将SystemRoutine的运行给出一个运行频率，相当于一个外部的人为的执行频率控制

SystemRoutine函数用于执行主要的逻辑，包含传感器（默认使用超声波模块）的数据收发，距离的检测，再有就是运行模式的控制，接着就是核心算法对环境进行整理之后对路径指导后运行电机进行小车的路径搜寻，小车运行运行完之后对于寄存器需要进行更新

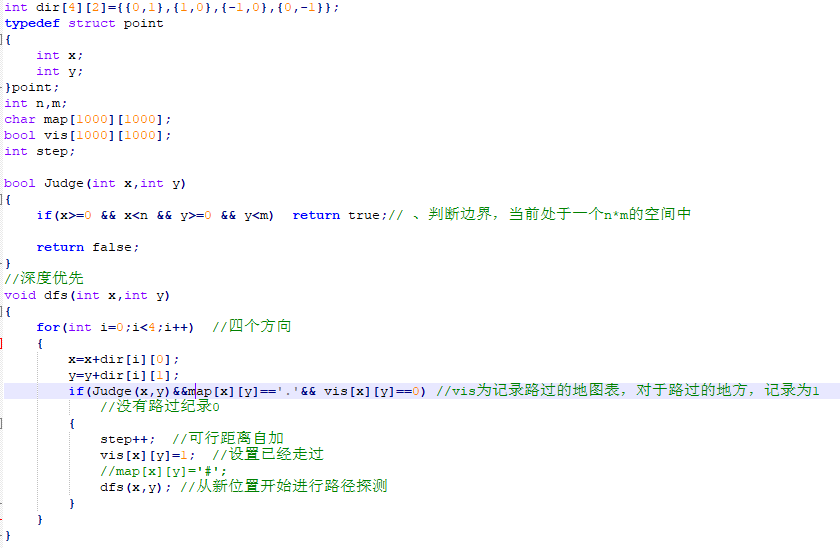


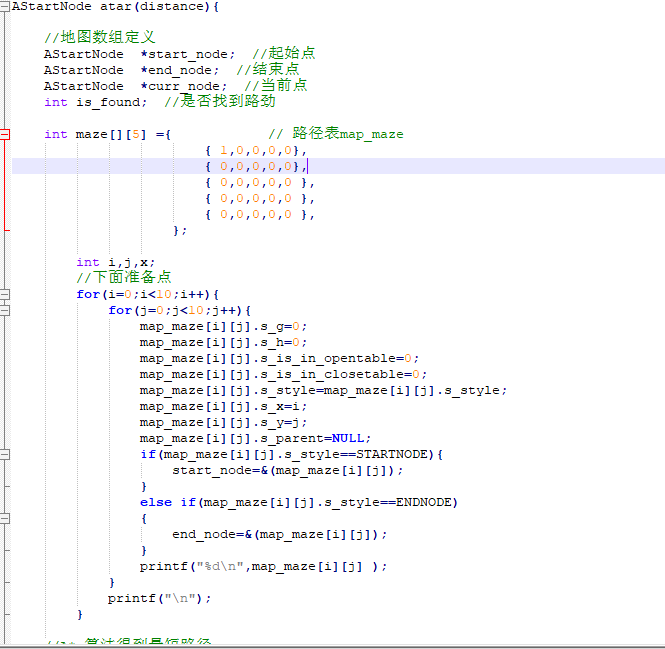
超声波避障模块代码以及注释

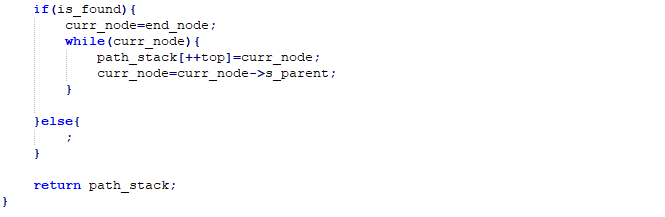
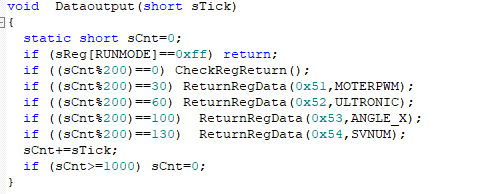




深度优先算法



A星算法

数据输出模块，用于将数数据输出到对应的串口以及电机上去。

## 第七部分:设备操作说明书

PCB板上丝印为“KEY”的红色按键有两种使用情况：长按表示切换功能模式，短按表示启动或者停机。这里需要注意的是，把开关拨到“ON”时，只需短按，方可进入0模式，也就是“超声波避障模式”，运行astar算法另外运行深度优先需要重新烧录，也可直接长按进入1模式，也就是“超声波避障”，dfs寻路。短按看停止运行。